Estágio Científico e Tecnológico I - EE015

Relatório

Predição de Séries Temporais Baseada em Redes Neurais Artificiais

Submetido à
Faculdade de Engenharia Elétrica e Computação (FEEC)

Departamento de Engenharia de Computação e Automação Industrial (DCA)
Faculdade de Engenharia Elétrica e de Computação (FEEC)
Universidade Estadual de Campinas (UNICAMP)
CEP 13083-852, Campinas - SP

Aluno: João Pedro de Oliveira Pagnan
Orientador: Prof. Levy Boccato

Campinas, 21 de junho de 2021

1 Introdução

A predição de séries temporais é uma das aplicações mais interessantes do tratamento de informação. O desafio de antecipar padrões de comportamento e construir modelos que sejam apropriados para explicar determinados fenômenos da natureza tem importância para a biologia, economia, automação industrial, meteorologia e diversas outras áreas da ciência [1].

É possível definir uma série temporal como sendo um conjunto de medidas feitas ao decorrer de um intervalo de tempo, podendo este ser contínuo ou discreto, sobre um fenômeno de interesse. Os sistemas cujas medições formam uma série temporal podem ser originados por processos determinísticos ou estocásticos [1].

Através da análise e interpretação de uma série temporal, podemos estimar os seus valores futuros, aumentando a informação que podemos obter das observações que já foram realizadas em um sistema.

Na literatura, encontramos diversos tipos de modelos para a predição de séries temporais, desde métodos clássicos lineares, como o modelo autorregressivo (AR) [1] até métodos não-lineares utilizando, por exemplo, redes neurais artificiais, sendo que dessas se destacam as redes do tipo *Multilayer Perceptron* (MLP) e as redes recorrentes, especialmente a *Long Short-Term Memory* (LSTM) [2] e a *Echo State Network* (ESN) [3].

Uma classe de sistemas dinâmicos particularmente relevante dentro do contexto de modelagem e predição de séries temporais está ligada à ideia de dinâmica caótica. Diversos fenômenos naturais, como a dinâmica populacional de uma espécie, a dinâmica atmosférica de uma região, ou até mesmo as órbitas de um sistema com três ou mais corpos celestes podem exibir comportamento caótico. Apesar de serem determinísticos (e, portanto, previsíveis), esses sistemas são extremamente sensíveis às condições iniciais [4]. Isso causa um problema para a predição das séries temporais originadas por eles, pois uma pequena incerteza na medida afetará toda a previsão.

Tendo em vista o desempenho de modelos não-lineares para previsão de diversas séries temporais [2], optamos por estudar a aplicabilidade de redes neurais artificiais à previsão de séries relacionadas a sistemas com dinâmica caótica.

Essa primeira parte do projeto de iniciação científica teve como objetivo estudar a base teórica das redes neurais artificiais e de outros regressores lineares clássicos, assim como estudar os fundamentos de sistemas dinâmicos e de dinâmica caótica. Os principais modelos estudados, juntamente com uma breve exposição de modelos lineares básicos, são apresentados na Seção 2. Já na Seção 3, veremos alguns conceitos fundamentais e a caracterização de sistemas caóticos.

O estudo dirigido começou abordando uma revisão de tópicos de probabilidade, teoria da informação e estimação. Em seguida, foi vista a teoria de regressores e classificadores clássicos [5]. Depois disso, o estudo se dirigiu para as redes neurais artificiais MLP e recorrentes [6], para, por fim, concluir o aprendizado de preditores com uma breve exposição dos modelos autorregressivos (AR) e autorregressivos de médias móveis (ARMA) [1]. Com a teoria de predição solidificada, o foco mudou para

os fundamentos da teoria de sistemas com dinâmica caótica, utilizando como base as referências [4] e [7].

Paralelamente aos estudos realizados durante o projeto de iniciação científica, o aluno também pôde fortalecer sua formação em aprendizado de máquina através da disciplina de pós-graduação **IA048 - Aprendizado de Máquina**, cursada como aluno especial. Com efeito, algumas atividades práticas desta disciplina foram importantes por estimularem o desenvolvimento de alguns ensaios de aplicações de redes neurais artificiais, que serão apresentados na Seção 4.

Por fim, na Seção 5 são indicados os próximos passos deste projeto de iniciação científica visando sua conclusão ao final deste primeiro semestre de 2021.

2 Modelos de Predição

2.1 Modelos Lineares

Apesar desta pesquisa focar na aplicabilidade de modelos preditores não-lineares utilizando redes neurais artificais, é pertinente darmos uma introdução ao assunto através de modelos lineares para essa aplicação, já aproveitando o momento para apresentarmos alguns conceitos básicos de predição.

2.1.1 Modelo Autorregressivo (AR)

No modelo autorregressivo (AR, do inglês *autoregressive*) o valor da série para um instante de tempo n, denotado por $x(n)$, é dado pela combinação linear dos valores passados a partir do instante $n - L - (K - 1)$ até o instante $n - L$, onde L é o passo de predição (quantos instantes de tempo à frente pretende-se predizer o valor da série) e K é a ordem do modelo.

Portanto, podemos dizer que, no modelo AR o valor da série temporal num instante n é dado por [8]:

$$x(n) = a_1 \cdot x(n - L) + a_2 \cdot x(n - L - 1) + ... + a_K \cdot x(n - L - (K - 1)) + \eta(n) \qquad (1)$$

onde a_k, $k = 1, 2, ..., K$ são os coeficientes que ponderam as amostras nos instantes passados e $\eta(n)$ é o erro instantâneo do modelo preditor. Esse erro instantâneo é um ruído branco (do inglês, *white noise*), possuindo média nula e variância σ_n^2 constante [1].

É interessante mencionar que, se considerarmos $L = 1$, ou seja, se estivermos predizendo o valor da série num instante seguinte ao atual, podemos dizer que:

$$\sum_{k=0}^{K} w_k \cdot x(n - k) = \eta(n) \qquad (2)$$

sendo $w_0 = 1$ e $w_k = -a_k$ para $1 \leq k \leq K$.

Perceba que o lado esquerdo de (2) é uma soma de convolução em tempo discreto, portanto, podemos interpretar o modelo AR como um sistema linear e invariante com o tempo (LIT) [8].

2.1.2 Modelo Autorregressivo e de Médias Móveis (ARMA)

O modelo ARMA (do inglês, *auto-regressive moving-average*) também leva em consideração o valor do ruído branco nos instantes de tempo anteriores ao atual [1]:

$$x(n) = \sum_{k=1}^{K} a_k \cdot x(n - L - (k-1)) + \sum_{k=1}^{M} b_k \cdot \eta(n - L - (k-1)) \tag{3}$$

Para ser parametrizado, o modelo ARMA necessita de métodos iterativos e/ou heurísticos. Isso é devido ao fato de que não há soluções em forma fechada para obter os coeficientes b_k. Além disso, é válido mencionar que durante a otimização desses parâmetros, devemos nos atentar a estabilidade desse sistema, afinal, os erros podem se acumularem, levando a uma divergência na saída do preditor [1].

Apesar dos modelos lineares terem e ainda serem bastante utilizados para a predição de séries temporais, para determinadas situações a regra linear aplicada pelos modelos AR e ARMA não é suficiente para realizar uma predição com um erro aceitável em sistemas mais complexos.

Devido a isso, optamos por direcionar a análise para modelos não-lineares, nesse caso, utilizando redes neurais artificais para a predição. Veremos então como são esses tipos de preditores.

2.2 Modelos Não-lineares

Os modelos não-lineares estudados foram as famosas redes neurais artificiais.

As redes neurais artificiais são ferramentas computacionais cujas estruturas são inspiradas no funcionamento das redes neurais biológicas presentes em cérebros de animais desenvolvidos, em especial do ser humano. Podemos interpretar um neurônio (tanto biológico, quanto artificial) como uma unidade de processamento de informação [9].

Analogamente, uma rede neural artificial é uma estrutura formada por vários neurônios artificiais interconectados, a qual é capaz de processar estímulos (sinais) de entrada e de produzir respostas conforme a tarefa desejada. Existem alguns modelos matemáticos para o neurônio artificial, sendo o *perceptron* um dos mais usuais (vide Seção 2.1.1). Além disso, os neurônios podem ser organizados de diferentes maneiras para construir a arquitetura (ou topologia) da rede neural, a qual é tipicamente estruturada

em camadas. Por fim, os neurônios artificiais podem exibir uma estrutura interna que varia de acordo com a arquitetura desejada para a aplicação, como observaremos na Seção 2.2.2 onde são discutidos alguns exemplos de redes recorrentes.

Veremos então os dois principais modelos de redes que serão utilizados nessa pesquisa:

2.2.1 Redes *Multilayer Perceptron* (MLP)

Um dos modelos mais utilizados para representar um neurônio artificial, o *Perceptron* [10], é apresentado na Figura 1.

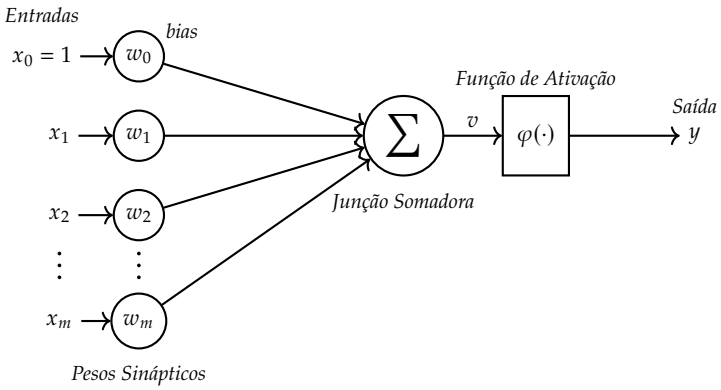

Figura 1: Modelo *Perceptron* para o neurônio artificial

Em termos matemáticos, a saída do neurônio pode ser escrita como:

$$y = \varphi(v) = \varphi\Big(\sum_{i=1}^{m} w_i x_i + w_0\Big) = \varphi\Big(\sum_{i=0}^{m} w_i x_i\Big) = \varphi(\mathbf{w}^T \cdot \mathbf{x}), \tag{4}$$

onde $\mathbf{w}$ é o vetor que contém os coeficientes, denominados de pesos sinápticos, que ponderam as entradas do neurônio.

A escolha da função de ativação $\varphi(\cdot)$ varia de acordo com a aplicação desejada. Ela pode ser desde uma função de *Heaviside*, a puramente linear $\varphi(x) = x$, ou até mesmo a tangente hiperbólica, a função logística ou outras funções não-lineares para mapeamentos mais complexos [6]. Na figura 2, vemos alguns exemplos de funções de ativação comumente utilizadas nos neurônios *Perceptron*, assim como as suas derivadas.

É interessante mencionar o fato de que, assim como podemos selecionar uma gama de funções não-lineares para a ativação do neurônio, de forma a realizar transformações não-lineares na entrada, também é possível utilizar funções de ativação lineares, ou seja, realizar transformações lineares assim como os modelos clássicos de predição (como o AR e o ARMA) fazem. Nesse caso, a falta da não-linearidade tornaria muito difícil ou até mesmo impossibilitaria que sistemas mais complexos fossem descritos de forma

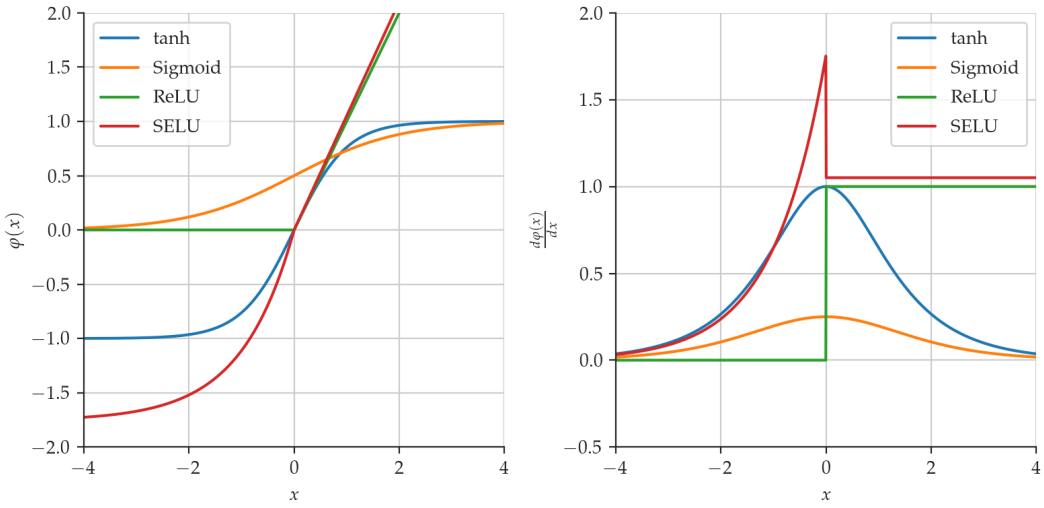

Figura 2: À esquerda, algumas funções de ativação comuns em tarefas de predição e regressão para o neurônio *Perceptron* e, à direita, suas derivadas

aceitável por redes neurais artificais desse tipo [11].

Tipicamente, uma rede neural MLP é composta por um número arbitrário N_L de camadas com n neurônios do tipo *Perceptron*, com a característica de que as saídas dos neurônios da l-ésima camada são propagadas para a frente, servindo como as entradas de todos os neurônios da camada seguinte $(l + 1)$. Esse processo é chamado de *feedforward*. Por isso, este tipo de rede é conhecida como totalmente conectada (ou densa). A figura 3 apresenta a estrutura típica das redes MLP.

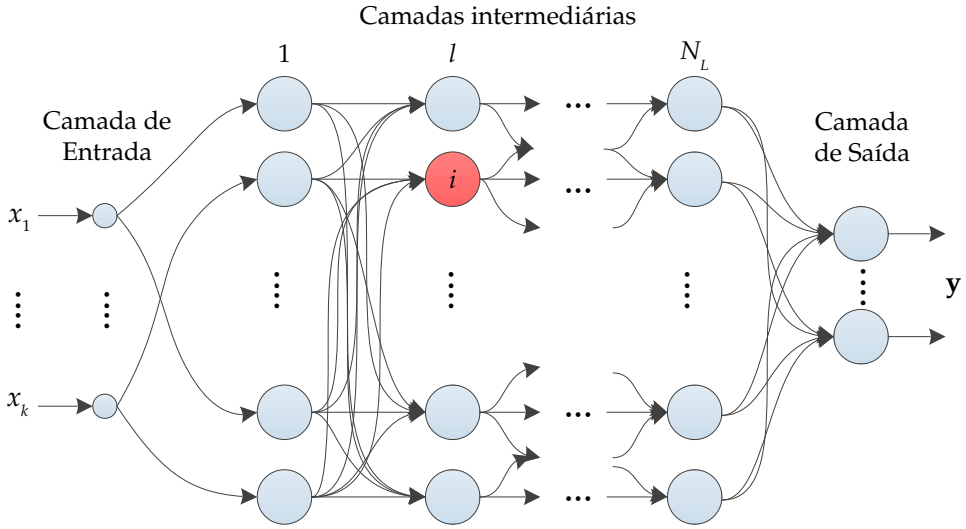

Figura 3: Estrutura típica de uma rede MLP (figura extraída de [12])

Pela figura vemos que, além das N_L camadas intermediárias com neurônios *Perceptron*, a estrutura das redes MLP contém uma camada de entrada, que possui ativação linear e apenas passa o atributo de entrada relacionado ao neurônio em específico (x_1, x_2, ..., x_k) para a primeira camada intermediária, e uma camada de saída que gera as respostas da rede neural para um vetor de entrada. A função de ativação da camada de

saída e o número de neurônios presente nela dependem da tarefa a ser realizada pela MLP. Por exemplo, em aplicações de regressão é comum o uso de um ou mais neurônios (dependendo se a saída do preditor será um vetor com um ou mais instantes de tempo) com função de ativação linear. Já em aplicações de classificação, a função de ativação pode variar entre a função logística (utilizada no caso binário em que queremos saber se uma entrada pertence ou não a uma classe) e entre a função *softmax* (que gera uma saída para cada classe, indicando a probabilidade de uma entrada pertencer à classe em específico) e, novamente, o número de neurônios de saída também varia com o tipo da classificação.

De forma similar a feita em (4), é possível representar a saída do i-ésimo neurônio da l-ésima camada intermediária, sendo que a anterior a esta possui n_{l-1} neurônios, a l-ésima possui n_l neurônios e a estrutura possui N_L camadas intermediárias, da seguinte forma:

$$y_i^l = \varphi^l\Big(\sum_{j=1}^{n_{l-1}} w_{ij}^l y_j^{l-1} + w_{i0}^l\Big) \tag{5}$$

onde w_{ij}^l representa o peso sináptico da conexão que liga o j-ésimo neurônio da camada $l-1$ ao i-ésimo neurônio da camada l, sendo que na primeira camada intermediária os sinais de entrada são os atributos do vetor de entrada, ou seja, $y_j^0 = x_j$ com $j = 1, ..., k$ [12].

Os pesos sinápticos $\mathbf{w}$ são ajustados com um processo iterativo de forma a minimizar uma função custo $J(\mathbf{w})$ que representa uma medida do erro entre as saídas geradas pela rede e as saídas desejadas (vale mencionar que $\mathbf{w}$ é um vetor com todos os parâmetros da rede). No caso de um problema de regressão ou predição, é comum que a função custo a ser minimizada seja o Erro Quadrático Médio (MSE, do inglês *Mean Squared Error*), assim, o problema envolve otimização não-linear irrestrita [9].

Para isso, é frequente o uso de algoritmos de otimização baseados em derivadas da função custo $J(\mathbf{w})$, como o método do gradiente descendente estocástico (SGD, do inglês *stochastic gradient descent*), o método de Nesterov (NAG, do inglês *Nesterov Accelerated Gradient*) e o algoritmo Adam (*Adaptive Moment Estimation*) [6]. O famoso algoritmo de retropropagação (*backpropagation*) do erro é empregado para viabilizar o cálculo das derivadas com relação aos pesos sinápticos dos neurônios situados nas camadas internas da rede.

Nesse caso, é comum dividirmos os métodos de otimização entre métodos de primeira ordem e de segunda ordem. Os métodos de primeira ordem utilizam as derivadas de primeira ordem da função custo, geralmente representadas na forma vetorial com o vetor gradiente:

$$\nabla J(\mathbf{w}) = \Big[\frac{\partial J(\mathbf{w})}{\partial w_1} \ \dots \ \frac{\partial J(\mathbf{w})}{\partial w_n}\Big]^T \tag{6}$$

assim, ao caminhar na direção contrária à apontada pelo vetor $\nabla J(\mathbf{w})$ obtemos, de forma iterativa, a minimização desejada. Logo, a regra de atualização dos pesos pode ser dada pela forma básica:

$$\mathbf{w}[k+1] \leftarrow \mathbf{w}[k] - \eta \nabla J\big[\mathbf{w}[k]\big] \tag{7}$$

Os algoritmos de otimização mencionados (SGD, Adam, NAG) utilizam variações da regra de atualização de **w** indicada em (7).

Para esta pesquisa, optamos por utilizar o algoritmo Nadam (*Nesterov Adaptive Moment Estimation*) [13].

É importante mencionar que os métodos de otimização aqui mencionados são métodos de busca local, ou seja, têm convergência esperada para um mínimo local, que não necessariamente é o mínimo global da função custo para a aplicação. Dizemos também que esses pontos mínimos possuem "bacias de atração"que, para valores adequados da taxa de aprendizado η, atrai o vetor de parâmetros **w** [5].

Por fim, vale dizer que o processo de treinamento normalmente é realizado com sequências de vetores de entrada **x**, chamadas de *mini-batch*, e chamamos um período de treinamento de época (do inglês *epoch*) [6].

O grande apelo das redes *Multilayer Perceptron* é que elas tem a capacidade de aproximação universal, ou seja, são capazes de aproximar qualquer mapeamento contínuo num domínio compacto com um nível de erro arbitrariamente pequeno. Até mesmo uma MLP com uma única camada intermediária e camada de saída linear já possui esta capacidade [14].

2.2.2 Redes Recorrentes *Long Short-term Memory* (LSTM)

Diferentemente das redes MLP que são *feedforward*, ou seja, que não reutilizam a informação processada dos padrões anteriores para gerar a próxima saída, a ideia central das redes recorrentes é que elas têm estruturas computacionais que podem armazenar os estados anteriores dos neurônios, possuindo também portas não-lineares que regulam o fluxo de informação de entrada e de saída da célula computacional [9]. Uma representação possível de uma célula de uma rede recorrente pode ser vista na Figura 4. Note que a saída é realimentada (com um atraso temporal) para a entrada do próprio neurônio.

Redes Recorrentes *Long Short-term Memory* (LSTM) Em especial, as redes LSTM se mostram atraentes pela possibilidade de criar e explorar memórias de curto e de longo prazo. A estrutura da célula ou camada LSTM, assim como as equações presentes nela, são apresentadas na Figura 5.

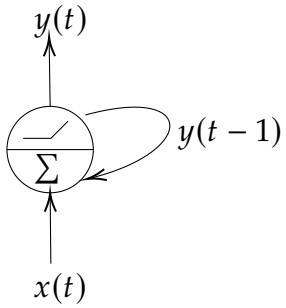

Figura 4: Célula da rede recorrente

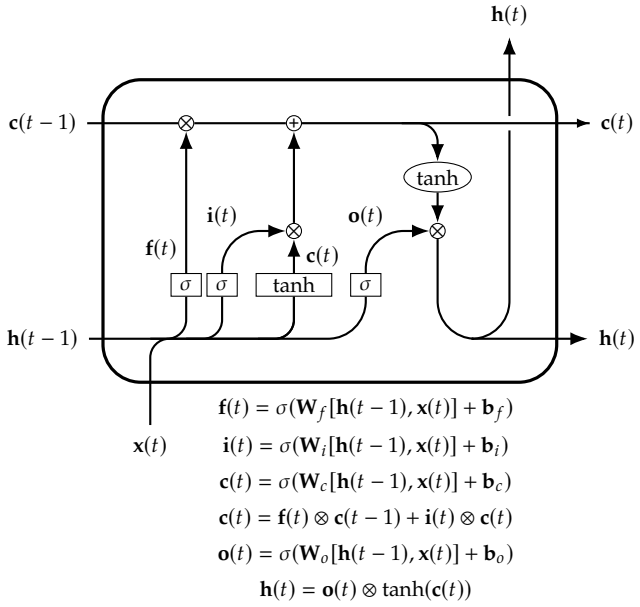

Figura 5: Estrutura e equações de uma célula/camada LSTM

As LSTMs manipulam o vetor $\mathbf{c}(t)$, aprendendo durante o treinamento o que deve ser guardado nele, o que deve ser descartado e o que deve ser aproveitado para gerar a saída $\mathbf{h}(t)$. Dessa forma, podemos dizer que a atualização do vetor de estados $\mathbf{c}(t)$ é feita com o descarte de informações e a incorporação de novidades vindas da entrada.

À semelhança das redes MLP, o treinamento de uma LSTM também é realizado através de algoritmos de otimização baseados em derivadas da função custo; a diferença é que agora é necessário propagar as derivadas ao longo da estrutura e, também, ao longo do tempo devido às realimentações. O algoritmo BPTT (*backpropagation-through-time*) representa a extensão do backpropagation para o cenário das redes recorrentes [6].

3 Sistemas com Dinâmica Caótica

Como dito anteriormente, sistemas com dinâmica caótica se destacam pois, apesar de serem determinísticos, apresentam dependência sensitiva em relação às condições iniciais (DSCI). Dessa forma, duas trajetórias que partem de posições relativamente próximas no espaço de estados podem evoluir de uma forma totalmente distinta devido às não-linearidades presentes que amplificam as diferenças entre essas condições iniciais [4].

De forma resumida, a dinâmica caótica é marcada pela presença dos seguintes aspectos [7]:

1. Forte sensibilidade com respeito às condições iniciais;

2. A evolução temporal das variáveis de estado (parâmetros de ordem do sistema) é rápida e tem uma aparência errática;
3. Um sinal originado por um sistema caótico tem espectro de potências contínuo e de faixa larga;
4. Há uma produção de informação por parte do sistema;
5. Dão origem a atratores estranhos (estruturas topológicas que ditam a evolução temporal do fluxo de um sistema caótico) [15].

4 Ensaios de aplicações de Aprendizado de Máquina

Como já foi mencionado, para complementar os estudos da iniciação científica, o aluno cursou como estudante especial a disciplina de pós-graduação **IA048 - Aprendizado de Máquina** da FEEC. O objetivo foi formar uma base sólida para o uso não só de redes neurais MLP, LSTM e regressores, como também noções de probabilidade, teoria da informação, classificadores, árvores de decisão, clusterizadores e outras ferramentas de *Machine Learning*.

Nessa matéria foram estudados os tópicos de probabilidade, estimação e teoria da informação, conceitos gerais de aprendizado de máquina, regressão linear, classificação linear, redes neurais artificiais MLP, recorrentes e convolucionais, *deep learning*, máquinas de vetores-suporte, aprendizado não-supervisionado, clusterização, modelos de mistura e extração de variáveis latentes, comitês de máquinas, árvores de decisão e *random forest* e aprendizado por reforço, utilizando materiais como [6, 9, 16, 5].

Algumas das atividades notórias desenvolvidas na disciplina foram o uso de modelos clássicos de regressores lineares e não-lineares para a predição de uma série temporal do número de manchas solares, testes com classificadores utilizando redes neurais MLP e redes convolucionais para cenários binários e multiclasse, e o desenvolvimento de um projeto final composto por um *Autoencoder* utilizando redes convolucionais e profundas para a filtragem de sinais.

O aluno concluiu a disciplina com uma média final de 9.6, alcançando o conceito A (máximo).

Com isso, além de reforçar a base teórica necessária para essa pesquisa, foi obtida uma prática com a programação, análise, teste, otimização e implementação de algoritmos de redes neurais, a qual será bastante útil para a sequência deste projeto de iniciação científica.

5 Próximos Passos

Como nessa primeira parte da iniciação o foco foi uma pesquisa bibliográfica dos temas a serem estudados nela, a segunda metade será voltada para a aplicação em si

da predição das séries temporais de sistemas caóticos.

Primeiramente, serão definidas as séries temporais que farão parte dos experimentos computacionais, buscando criar cenários diversificados para a análise do comportamento das redes neurais. Algumas possibilidades para o trabalho são os dados referentes ao mapa logístico [17] e ao mapa de Hénon [18], a famosa série Mackey-Glass [19] e dados de dinâmica populacional de uma espécie.

Em seguida, determinaremos aspectos mais fundamentais das redes neurais que serão utilizadas, como, por exemplo, a arquitetura empregada, assim como as métricas para o treinamento e análise.

Após isso, faremos a aplicação das redes neurais à predição das séries escolhidas, avaliando a sensibilidade paramétrica de cada estrutura na busca das melhores configurações, a fim de traçar um quadro comparativo entre as técnicas consideradas.

Por fim, compilaremos os resultados no relatório final, de forma a conter uma discussão ampla e representativa dos ensaios realizados e das conclusões obtidas.

Referências

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] J. T. Connor, R. D. Martin, and L. E. Atlas, "Recurrent neural networks and robust time series prediction," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 240–254, 1994.

[3] H. Jaeger, "Echo state network," *scholarpedia*, vol. 2, no. 9, p. 2330, 2007.

[4] N. Fiedler-Ferrara and C. P. C. do Prado, *Caos: uma introdução*. Editora Blucher, 1994.

[5] T. Hastie, R. Tibshirani, and J. Friedman, *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[6] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.

[7] R. R. de Faissol Attux, "Sobre dinâmica caótica e convergência em algoritmos de equalização autodidata," dissertação (mestrado), Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2001.

[8] S. S. Haykin, *Adaptive filter theory*. Pearson Education India, 2008.

[9] S. Haykin, *Neural networks and learning machines, 3/E*. Pearson Education India, 2010.

[10] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[11] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.

[12] L. Boccato *et al.*, *Novas propostas e aplicações de redes neurais com estados de eco*. Tese (doutorado), Universidade Estadual de Campinas, Faculdade de Engenharia Elétrica e de Computação, Campinas, SP, 2013.

[13] T. Dozat, "Incorporating nesterov momentum into adam," 2016.

[14] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[15] D. Ruelle and F. Takens, "On the nature of turbulence," *Les rencontres physiciens-mathématiciens de Strasbourg-RCP25*, vol. 12, pp. 1–44, 1971.

[16] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[17] R. M. May, "Simple mathematical models with very complicated dynamics," *The Theory of Chaotic Attractors*, pp. 85–93, 2004.

[18] M. Hénon, "A two-dimensional mapping with a strange attractor," in *The Theory of Chaotic Attractors*, pp. 94–102, Springer, 1976.

[19] M. C. Mackey and L. Glass, "Oscillation and chaos in physiological control systems," *Science*, vol. 197, no. 4300, pp. 287–289, 1977.